固有値計算のための高性能精度保証ライブラリの開発：最新成果と自動チューニング機能

片桐 孝洋

名古屋大学情報基盤センター

第13回 自動チューニング技術の現状と応用に関するシンポジウム（ATTA2021）
2021年12月13日（月）15:15-15:45

2021年度 JHPCN採択課題(国際課題)

- タイトル
  - Developing Accuracy Assured High Performance Numerical Libraries for Eigenproblems
- 期間
  - 2021年4月～2022年3月 (1年間)
  - 3年計画の3年目 (最終年度)
- 構成
  - 副代表：
    - Weichung Wang　(National Taiwan U.),
      Takeshi Ogita (Tokyo Women's U.)
  - 主な構成員：
    - K. Nakajima (U. of Tokyo), S. Ohshima (Nagoya U.),
      K. Ozaki (Shibaura I.T.), O. Marques (LBNL),
      F-N. Hwang (National Central U.)

Aim of This Study

To establish accuracy assured numerical computing, we focus on the following three topics:

1. Developing an accuracy assured numerical libraries for eigenproblems;
2. Development of high-performance implementation and auto-tuning technology for the developed accuracy assured numerical library;
3. Discussing an extension to non-liner problems based on obtained knowledge of accuracy assured algorithms.

(1) Interdisciplinary Research

Researchers for Accuracy Assurance Problems

Researchers for High-performance Computing

The main target of 2021 FY
(3) Developing Auto-tuning Function

(2) Assured algorithm

Ogita-Aishima's Iteration for Accurate Eigenpairs (Quadratic Convergence)

$$
\begin{aligned}
&1:\ R \leftarrow I - \widehat{X}^T \widehat{X} \\
&2:\ S \leftarrow \widehat{X}^T A \widehat{X} \\
&3:\ \tilde{\lambda}_i \leftarrow \frac{s_{ii}}{1 - r_{ii}} \quad (\text{for } 1 \le i \le l) \\
&4:\ \tilde{D} \leftarrow \mathtt{diag}(\tilde{\lambda}) \\
&5:\ \omega \leftarrow 2(\|S - \tilde{D}\|_2 + \|A\|_2 \|R\|_2) \\
&6:\ \tilde{e}_{ij} \leftarrow \begin{cases} \dfrac{s_{ij} + \tilde{\lambda}_j r_{ij}}{\tilde{\lambda}_j - \tilde{\lambda}_i} & (|\tilde{\lambda}_i - \tilde{\lambda}_j| > \omega) \\ r_{ij}/2 & (\mathtt{otherwise}) \end{cases} \\
&\qquad (\text{for } 1 \le i, j \le l) \\
&7:\ \tilde{X} \leftarrow \widehat{X} + \widehat{X}\tilde{E}
\end{aligned}
$$

A Result from 2020 FY

$$Ax=\lambda x,\ AX=XD$$

A: real symmetric matrix (binary64)

X: eigenvector matrix, D: eigenvalue matrix

Supercomputer "Flow" (Type I)

(matrix dimension: 2^{16}, 64 nodes)

pdsyevd: 240 sec, Iteration: 400 sec: Ratio: 1.66

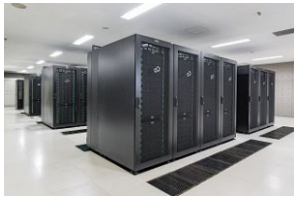
Oakforest-PACS

(matrix dimension: 2^{16}, 64 nodes)

pdsyevd: 880 sec, Iteration: 2120 sec: Ratio: 2.40

Lower is better.

Adaptation of Supercomputers



Accurate matrix multiplication is performed by Ozaki Method with six matrix multiplications.

(Ref. 内野 佑基, 尾崎 克久, 荻田 武史: "実対称行列の固有値分解に対する反復改良法の大規模並列環境における実装と評価", 第177回ハイパフォーマンスコンピューティング研究発表会, Dec. 2020)

VNC (Verified Numerical Computation)-HPC ライブラリ

- ポスト「京」萌芽的課題1：基礎科学のフロンティアー極限への挑戦ー：「極限の探究に資する精度保証付き数値計算学の展開と超高性能計算環境の創成」（代表：荻田武史 東京女子大学教授）～2019年（終了）で開発
- 精度が保証された計算結果を実用的に得られるような超高性能計算環境を構築
- 以下のライブラリのソースコードを公開中

1. **OzBLAS: Accurate and Reproducible BLAS based on Ozaki scheme** (尾崎スキームに基づく高精度基本線形計算ライブラリ) [for PC, GPU]
2. **GEMMTC: GEMM using Tensor Cores** (Tensorコアを用いた行列積プログラム) [for GPU]
3. **DHPMM_F for GPU: High-precision Matrix Multiplication with Faithful Rounding** (高精度行列積プログラム) [for GPU]
4. **PDDOTK: K-fold Precision Dot Product** (分散並列版高精度内積計算プログラム) [for PC, FX100]
5. **BLAS-DOT2: Higher-precision BLAS based on Dot2** (高精度内積計算アルゴリズムDot2に基づく基本線形計算ライブラリ) [for GPU]
6. **LINSYS_VR: Verified Solution of Linear Systems with Directed Rounding** (連立一次方程式に対する精度保証プログラム（丸めモードの変更機能付き）) [for K Computer, FX100]
7. **LINSYS_V: Verified Solution of Linear Systems** (連立一次方程式に対する精度保証プログラム) [for PC, K Computer, FX100]
8. **DHPMM_F: High-precision Matrix Multiplication with Faithful Rounding** (高精度行列積プログラム) [for PC, K Computer, FX100]

HP：http://www.math.twcu.ac.jp/ogita/post-k/index.html

名古屋大学 大学院情報学研究科 M1　青木 将太
名古屋大学情報基盤センター 准教授　大島聡史

精度保証ライブラリの自動チューニング（FY2021の成果）

第13回 自動チューニング技術の現状と応用に関するシンポジウム

名古屋大学
NAGOYA UNIVERSITY

発表の流れ

- 背景
- 機械学習モデルの特徴量解析
- 高精度行列-行列積アルゴリズム（尾崎の方法）
- 予備評価
- おわりに

発表の流れ

- 背景
- 機械学習モデルの特徴量解析
- 高精度行列-行列積アルゴリズム（尾崎の方法）
- 予備評価
- おわりに

背景

- 人工知能(AI)が出力する答えを検証が不十分なまま利用することで様々な社会的な問題が生じている
- AI出力に対して人間による検証が必須
- 性能チューニング工数の削減を目的に、ソフトウェア自動チューニング（Software Auto-tuning, AT）技術が研究されている
  - AIのAT機構への適用が進んでいる

    →精度保証数値計算ライブラリの性能パラメタチューニングにAIを適用した事例の＜AI出力結果の説明性＞ を検証

    →Scientific XAI (SXAI)

名古屋大学 NAGOYA UNIVERSITY

発表の流れ

- 背景
- 機械学習モデルの特徴量解析
- 高精度行列-行列積アルゴリズム（尾崎の方法）
- 予備評価
- おわりに

説明可能なAI（XAI）

- 機械学習が出した結果が説明できるか？
  - Explainable AI (XAI)
- 説明可能なAIには以下の２種がある[1]
  - 説明可能性
    - 予測の判断理由を人間が理解できるように説明する技術
    - 例）LIME、SHAPなどのツール
  - 解釈可能性
    - 内部構造を解析することで、予測に至る計算過程を確認できるようなAI技術
    - 例）決定木を作る

→ここでは＜説明可能性＞を扱う

[1] 大坪ほか：「XAI(説明可能なAI)：そのとき人工知能はどう考えたのか」、リックテレコム、2021

大局説明と局所説明

- 大局説明 (Global Explanations)
  - 「AIモデルの全体的なふるまい」を理解する[1]
  - 例）SHAPなどのツール
- 局所説明 (Local Explanations)
  - 「個々の予測結果の判断理由」を理解する[1]
  - 例）LIMEなどのツール

→　ここでは、数値計算処理のATに適用したAIについて、双方の説明性を取り扱う

[1] 大坪ほか：「XAI(説明可能なAI)：そのとき人工知能はどう考えたのか」、リックテレコム、2021

LIME (Local Interpretable Model-agnostic explanations) [2]

- Local surrogate model
  - ブラックボックスモデルの個々の予測を説明するために用いられる解釈可能なモデル
- 理由を人間でも理解できるように提示
- 各特徴がどの程度分類に貢献しているか調べ分類器の予測結果を説明
  - 説明対象データを摂動させた近傍データからAIモデルを得る
- 任意の分類器に適用できる
- 欠点
  1. 毎回、説明がぶれる（乱数利用）
  2. ハイパーパラメタ調整がある
  3. 説明できないケースがある（→あきらめて他の方法を利用）

[2] M. T. Ribeiro, S. Singh, and C. Guestrin: Why should I trust you?: Explaining the predictions of any classifier, Proc. of 22nd ACM SIGKDD, pp.1135-1144, 2016.

SHAP（SHapley Additive exPlanations）[3]

- 協力ゲーム理論のシャープレイ値（Shapley Value）を機械学習に応用
  - それなりの妥当性がある
- 近似的にシャープレイ値を算出
  - ツリー系アンサンブルモデル：高速で正確なシャープレイ値の算出
  - ディープラーニングモデル：高速で近似的なシャープレイ値の算出
  - 一般的なアルゴリズム：シャープレイ値の推定値の算出
- 欠点
  1. 計算量が多い（→近似的な評価値を利用）

[2] S. Lundberg, S-I. Lee, A Unified Approach to Interpreting Model Predictions, 2017
https://arxiv.org/abs/1705.07874

発表の流れ

- 背景
- 機械学習モデルの特徴量解析
- 高精度行列-行列積アルゴリズム（尾崎の方法）
- 予備評価
- おわりに

Background, objective, and motivation of our work

- High-accuracy and low-accuracy calculations are one of the important calculation techniques which can solve large-scale and complicated calculations.
- Some studies have investigated the accuracy assurance of BLAS and LAPACK.
  - These studies have used mixed procedure computations and arbitrary digit computations.
  - Most of BLAS and LAPACK libraries tend to place less emphasis on the accuracy of the computation results.
- Why high-accuracy and assured calculations are not widespread?
- => TIME
- We consider to calculate it on GPU and shorten the time.
- This work will be helpful for large-scale and complicated calculations on current and future generation computers. (⇨ topics of interest of this workshop)

In particular, we will focus on the following topics of interest, but not limited to:

- Programming models, languages and frameworks for facilitating HPC software evolution and refactoring.
- Algorithms and implementation methodologies for future-generation computing systems, including manycores and accelerators (GPUs, Xeon Phi, etc).
- Automatic performance tuning techniques, runtime systems and domain-specific languages for hiding the complexity of underlying system architectures.
- Practices and experiences on porting of legacy applications and libraries.

High-precision matrix-matrix multiplication algorithm

- Our target calculation: assured matrix-matrix multiplication (MMM) method proposed by Ozaki et al.
  - hereinafter, we refer to this MMM method as the Ozaki method
- Overview of the Ozaki method:

consider $C = AB$

- A : a matrix of size m * l
- B : a matrix of size l * n
- C : a matrix of size m * n

Step1: error-free transformation

$$A = A^{(1)} + A^{(2)} + A^{(3)} + \ldots + A^{(p)}$$
$$B = B^{(1)} + B^{(2)} + B^{(3)} + \ldots + B^{(q)}$$

The elements in the matrices with lower indices are given with a higher number of digits.

Step2: individual MMM

$$AB = (A^{(1)} + A^{(2)} + \ldots + A^{(p)})(B^{(1)} + B^{(2)} + \ldots + B^{(q)})$$
$$= A^{(1)} B^{(1)} + A^{(1)} B^{(2)} + A^{(2)} B^{(1)} + \ldots + A^{(p)} B^{(q)}$$

Step3: Accurate Sum

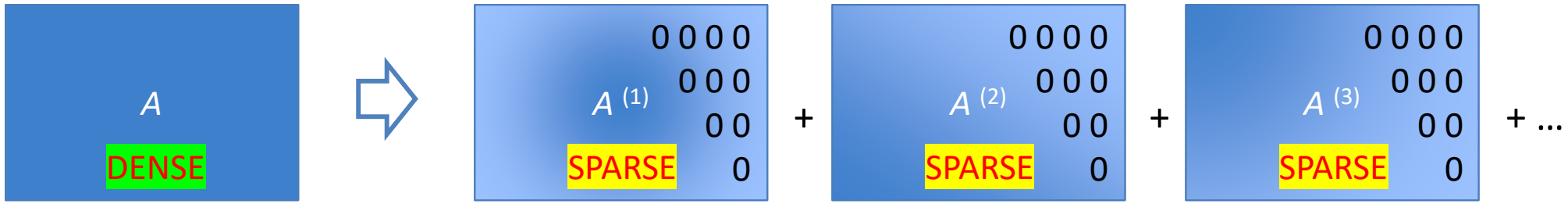
$fl(A^{(i)} B^{(j)}) = A^{(i)} B^{(j)}$ for $1 \leqq i \leq p,\ 1 \leqq j \leq q.$

$$= fl(A^{(1)} B^{(1)}) + fl(A^{(1)} B^{(2)}) + fl(A^{(2)} B^{(1)}) + \ldots + fl(A^{(p)} B^{(q)})$$
$$= C_1 + C_2 + \ldots + C_{pq}$$

fl is a floating-point arithmetic with rounding to the nearest

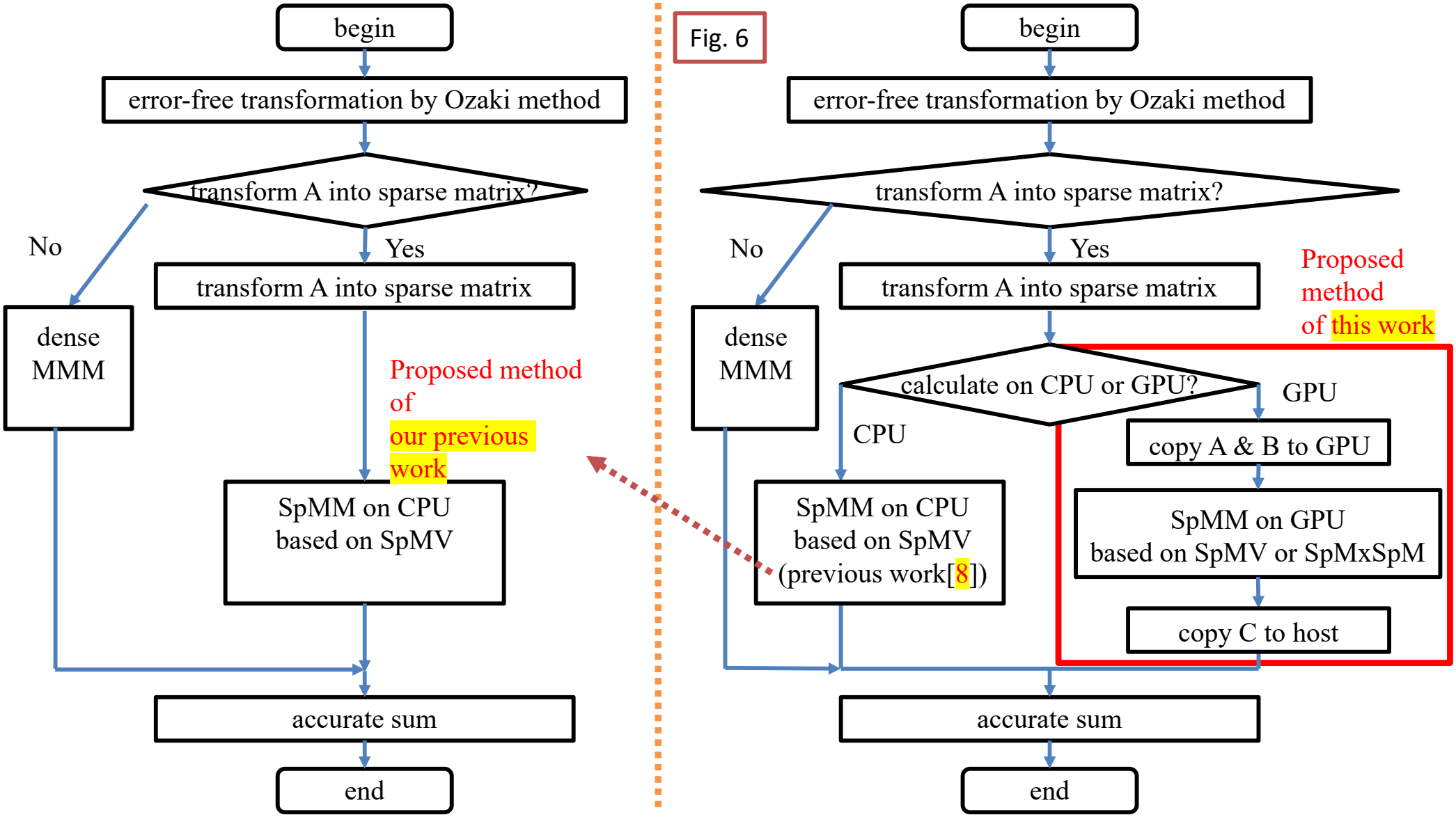
Previous work and Proposed method

- When the value ranges of the input matrix elements are large, error-free transformation generates many sparse matrices.

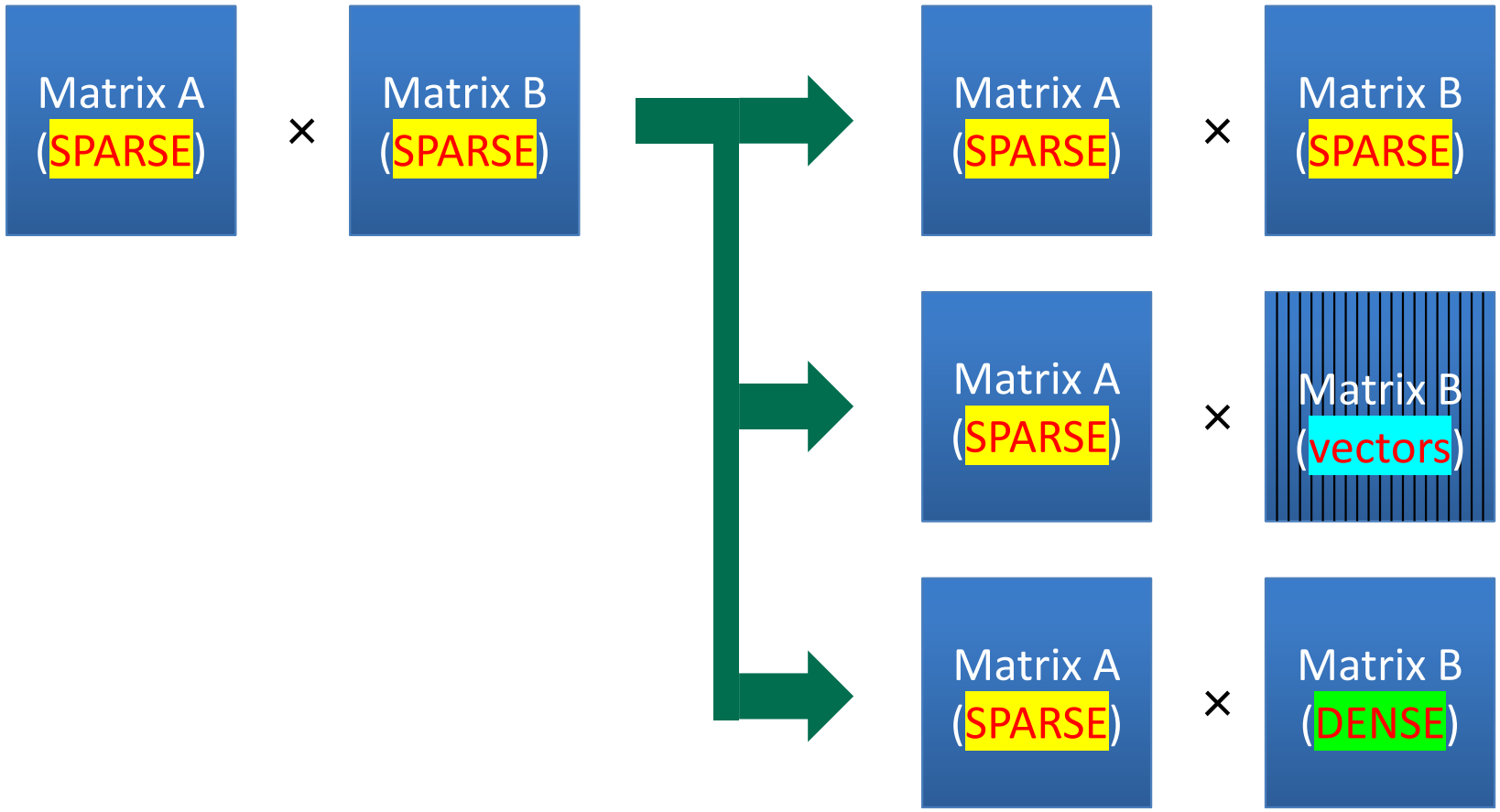


- In this case, many double precision general matrix - matrix multiplication (dgemm) are performed. Transforming dense matrices into sparse matrices and performing sparse matrix operations will require shorter calculation time than dense matrix operations.

- Therefore, our previous work proposed to transform the target matrices into sparse matrices and calculate sparse matrix computations on CPU.
  - Considering the performance, sparse matrix - vector multiplications (SpMV) are used.
- In this study, we propose to calculate these sparse matrix computations on GPU.


begin
error-free transformation by Ozaki method
transform A into sparse matrix?
No
Yes
dense MMM
transform A into sparse matrix
Proposed method of our previous work
SpMM on CPU based on SpMV
accurate sum
end
Fig. 6
begin
error-free transformation by Ozaki method
transform A into sparse matrix?
No
Yes
dense MMM
transform A into sparse matrix
Proposed method of this work
calculate on CPU or GPU?
CPU
GPU
SpMM on CPU based on SpMV (previous work[8])
copy A & B to GPU
SpMM on GPU based on SpMV or SpMxSpM
copy C to host
accurate sum
end

How to calculate Sparse Matrix A - Matrix B multiplication?



高精度行列積ライブラリ(尾崎の方法)の実装種別と表記

1. dgemmによる実装　1dgemm
2. CRS形式でのSpMV(内部並列)　2CRS
3. CRS形式でのSpMV(外部並列)　3CRS
4. CRS形式でのSpMV(複数右辺による内部並列化) 4CRS
5. CRS形式でのSpMV(複数右辺による内部並列化(ブロッキング))　5CRS
6. ELL形式でのSpMV(内部並列)　6ELL
7. ELL形式でのSpMV(外部並列)　7ELL
8. ELL形式でのSpMV(複数右辺による内部並列化）8ELL
9. ELL形式でのSpMV(複数右辺による内部並列化(ブロッキング))　9ELL
10. Batched BLASを適用した方式　10GPU
11. dgemmによる実装(GPU)　11GPU
12. CRS形式でのSpMV(GPU)　12GPU
13. ELL形式でのSpMV(GPU)　13GPU
14. CRS形式でのSpMM(GPU)　14GPU

発表の流れ

- 背景
- 機械学習モデルの特徴量解析
- 高精度行列-行列積アルゴリズム（尾崎の方法）
- 予備評価
- おわりに

評価環境

- 計算機：名古屋大学情報基盤センター設置の**スーパーコンピュータ「不老」Type Ⅱ サブシステム**
  - CPU
    - Intel Xeon Gold 6230, 20コア, 2.10 - 3.90 GHz × 2 ソケット
  - GPU：機械学習モデル生成用
    - NVIDIA Tesla V100 (Volta) SXM2, 2,560 FP64コア, upto 1,530 MHz × 4 ソケット
- LIMEは ver. 0.2.0.1 を利用
- SHAPは ver.0.39.0 を利用
- 機械学習モデル（分類器）
  - scikit-learn ver. 0.24.1のランダムフォレスト
  - 尾崎の方法の実装選択中の11種類

テスト行列生成方法

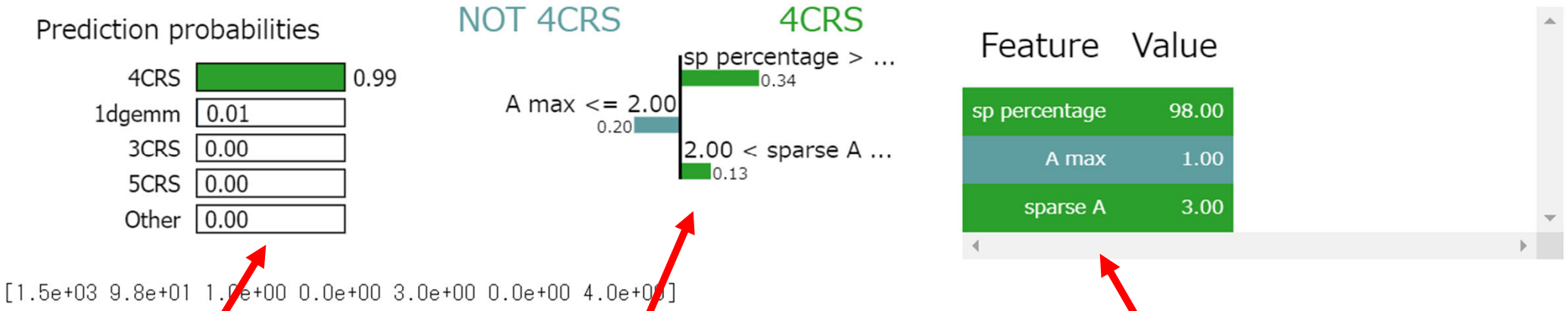
- 行列１：行列 A, B の要素を 0~1 の範囲で生成し，ある疎度分の要素に対しpow(10,rand( )%Φ) で生成した値を挿入
  - Φ=30まで
    (Φが一定数を超えるとpythonで扱える範囲を超える)
- 行列２：行列 A, B を単位行列とある疎度分の要素に対して 0~1 の範囲で生成
  - 疎度90から98
- 行列３：行列A，Bを単位行列とある疎度分の要素に対してpow(10,rand()%Φ)で生成した値を挿入
  - Φ=30まで
- 乱数行列：以上の**乱数の種は固定**

学習データの数と予測精度

- 学習データ
  1. 行列１：大きな値が作れる、疎度がほぼ0
  2. 行列２：0～1の範囲の疎行列
  3. 行列３：大きな値が作れる、任意の疎度
  - 行列サイズ1000～4000
- 機械学習情報
  - 方法：ランダムフォレスト
  - 説明変数：７変数
    - １）行列サイズ、２）入力行列の疎度、３）Aの要素最大値、４）Aの要素最小値、５）A疎な分解数、６）A密な分解数、７）Bの分解数
  - 学習データ数：199個
  - テストデータ数：23個
  - 正答率：91.3%
- ※ここでは、汎用的で予測精度の高い分類器を作るのが目的ではないので、精度にはこだわらない

LIMEによる出力例

●あるケースによる予測結果を説明（局所説明）



● 予想される目的変数の確率を表示

「4CRS」が99%予測される

● LIMEによる説明変数の寄与度を表示

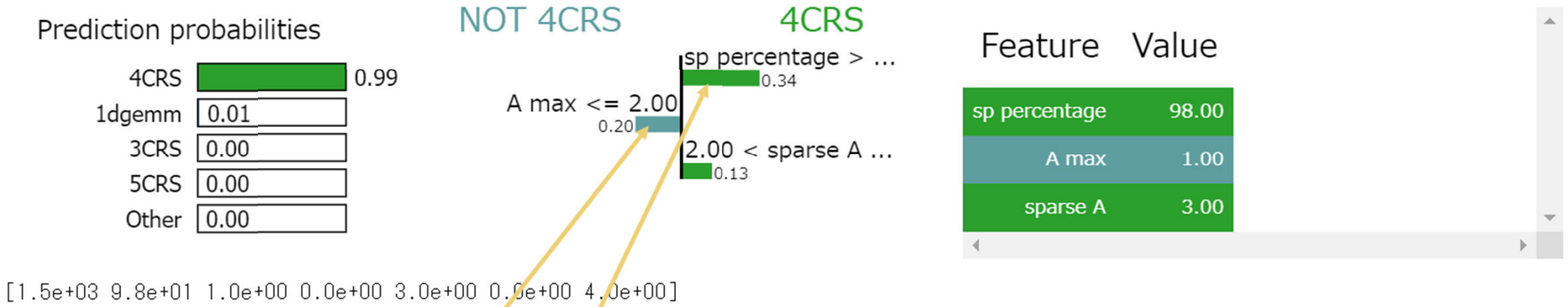
「4CRS」を予測する場合、「sp percentage」の寄与（正）が34%

● このケースの、説明変数の値を表示

名古屋大学 NAGOYA UNIVERSITY

LIMEによる分析結果（1／3）

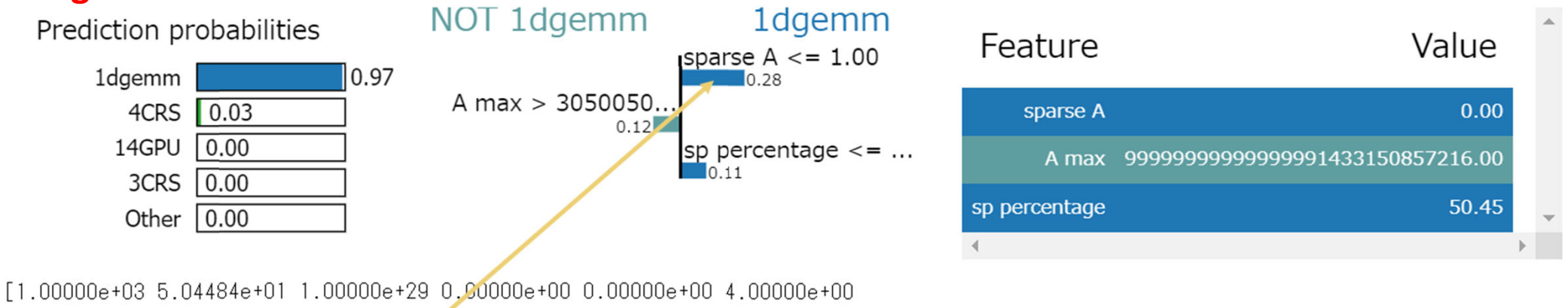
4CRSを99%予測



- 入力行列の疎度が高い（sp percentage）と4CRSと予想される
  →取得データ・数値アルゴリズム的に、正しい判断
- Aの要素の最大値が小さいと別の実装が速い可能性がある
  →A maxが負の要因となってもおかしくない

LIMEによる分析結果（2／3）

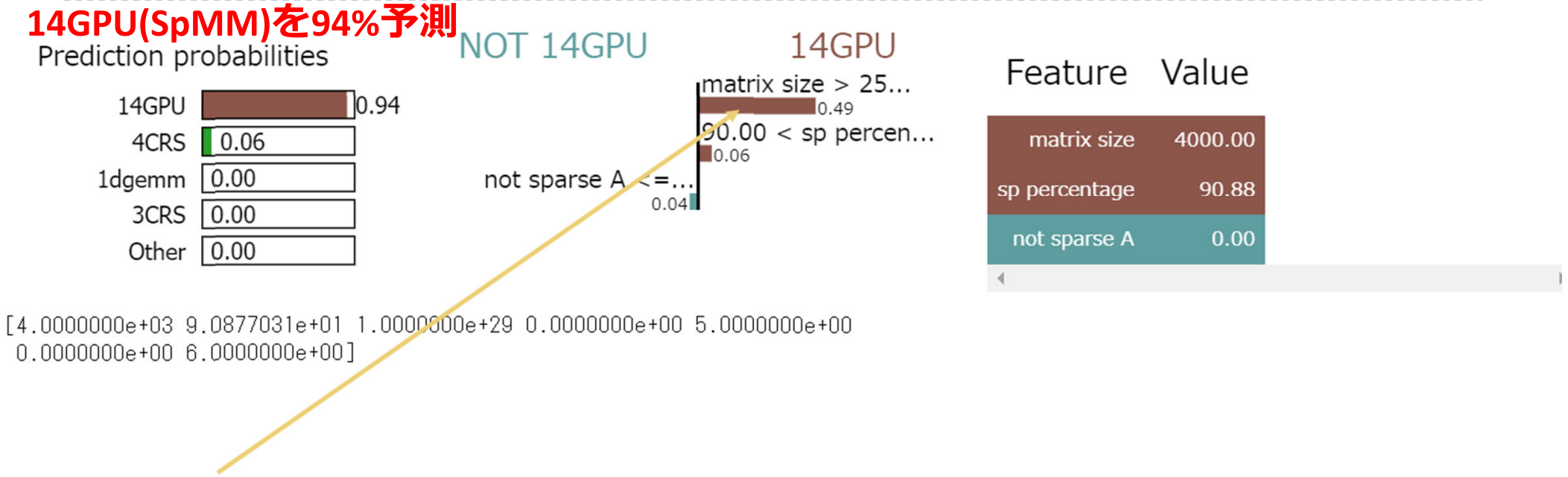
1dgemmを97%予測



- 全行列が密行列の時の計算：CPUのdgemmか、GPUのdgemmのみ
- 今回の学習データ：CPUのdgemmが速い

→分解行列の疎行列の数（sparse A）が0の場合、1dgemmを予測する要因として正しい

LIMEによる分析結果（3／3）



●学習データ：行列サイズが大きいと14GPU（CRS形式でのSpMM(GPU)）が速い傾向がある

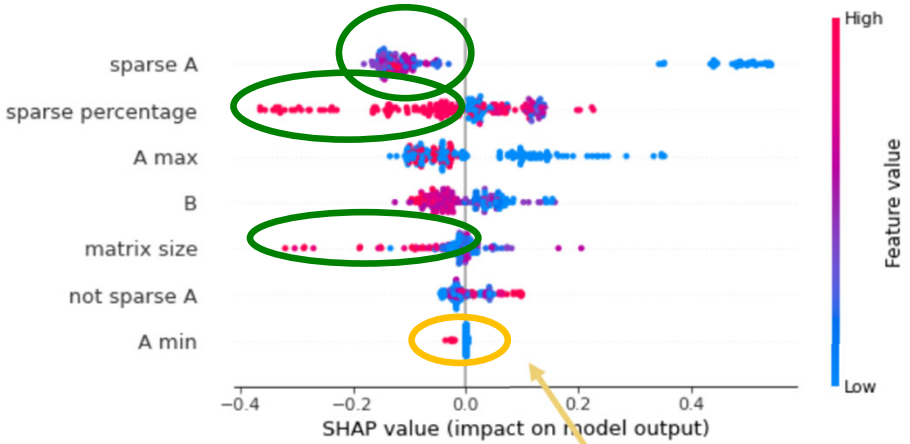
→（この行列：乱数行列では）

14GPUの予測の解釈として正しい

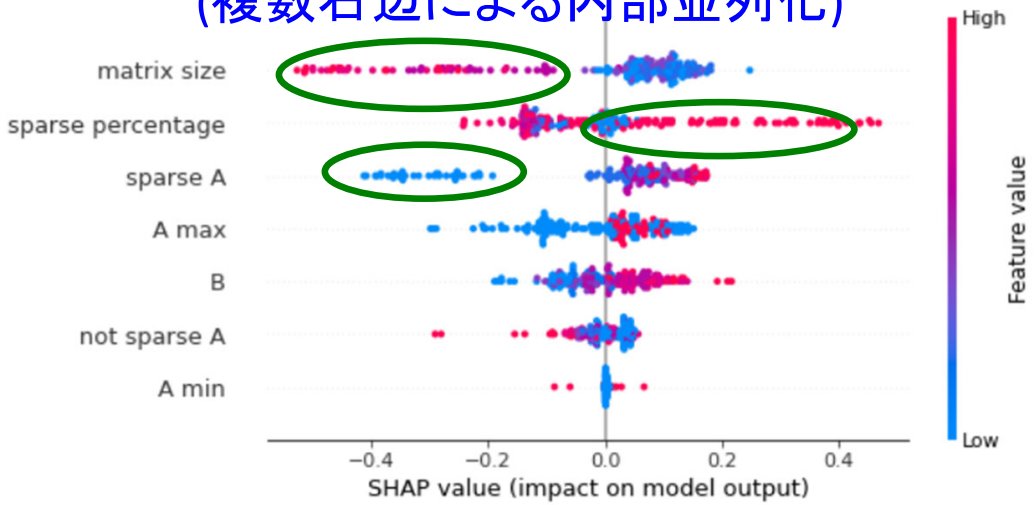
SHAPによる分析結果

- 「SHAP値の絶対値が大きく（要因が大きい）、色が近く（同じような説明変数の値）、塊が大きい、もしくは並んでいる（ケース数が多い）」 箇所に注目

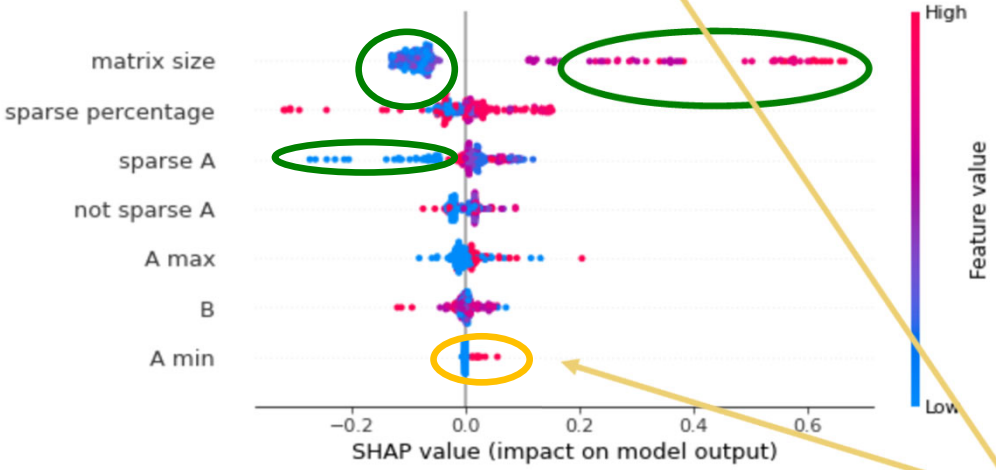
(1) 予想：dgemm



(2) 予想 ： CRS形式でのSpMV
(複数右辺による内部並列化)



(3)予想 ： CRS形式でのSpMM(GPU)



- (1) dgemm予想：Aの疎な分解数(－)、入力行列の疎度(－：値が大)、行列サイズ(－：値が大)の要因 →妥当
- (2) CRS（SpMV）予想：行列サイズ(－：値が大)、入力行列の疎度(＋：値が大)、Aの疎な分解数（－：値が小）の要因 →妥当
- (3)CRS（SpMM）（GPU）予想：行列サイズ（－：値が小、＋：値が大）、Aの疎な分解数（－：値が小）の要因 →妥当
- **Aの要素の最小値は、ほとんどの場合0**
  →予想に影響せず、良い説明変数ではない
  ←指摘されて分かる事項（この行列の場合には）

まとめ（XAI）

- 高精度行列積ライブラリの実装選択ATに機械学習を適用した事例で、説明可能なAI（XAI）ツールを利用しAI出力結果を分析
- 局所説明ツールLIME、大局説明ツールSHAP、ともに、対象の特性を考慮し、妥当な分析結果が得られることを確認した

今後の課題

1. XAIを活用したAT方式の開発
   - 自動で、寄与の大きな説明変数を学習を強化する
   - 寄与が少ないなら説明変数から外して、AT時間を短縮
2. XAI環境の高速化・高度化
   - 特にSHAPでフレームワーク(scikit-learn)の疎行列演算の高速化
   - 1のAT連結：説明結果から学習を（なんども）やり直す
     →機械学習のパラメタ並列実行・分散並列高速化が必須
     →スパコンでのXAI環境の整備　→AT機構の高度化